Claims

1. (Original) A method comprising:

selectively activating and deactivating particular simulation domains in a simulation environment such that a resolution and a performance for a circuit design being simulated is dynamically modified; and

said simulation environment comprising a plurality of simulation domains.

2. (Original) The method of claim 1 wherein the plurality of simulation domains comprises at least one of a software execution domain, a hardware simulation domain, and an abstract model simulation domain.

3. (Original) The method of claim 2 wherein the software execution domain comprises at least one of a native processor package, an instruction set simulator (ISS), and a programming language simulator to model software execution in one or more processors.

4. (Original) The method of claim 2 wherein the hardware simulation domain comprises at least one of a logic simulator and a programming language simulator.

5. (Original) The method of claim 4 wherein the logic simulator comprises one of a hardware description language (HDL) based simulator, a gate-level simulator, a simulation accelerator, a system simulator, a cycle simulator, and a programmable hardware emulator.

6. (Original) The method of claim 4 wherein the programming language simulator comprises at least one of a C programming language simulator, a C++ programming language simulator, a simulator using a C-based language, a simulator using a C++ based language, and a JAVA programming language simulator.

7. (Original) The method of claim 1 wherein each of the plurality of simulation domains comprises at least one model of a circuit element in the circuit design.

8. (Original) The method of claim 1 further comprising:

partitioning the circuit design into the plurality of simulation domains based on a partition criteria.

9. (Original) The method of claim 8 wherein the partition criteria comprises at least one of an abstraction level, a simulation type, and a function type.

10. (Original) The method of claim 9 wherein partitioning the circuit design based on the abstraction level partitions the circuit design into at least one of a pin-level domain, a bus-level domain, and a transaction-level domain.

11. (Original) The method of claim 9 wherein partitioning the circuit design based on the simulation type partitions the circuit design into at least one of a software execution domain, a logic simulator domain, and a programming language simulator domain.

12. (Original) The method of claim 9 wherein partitioning the circuit design based on the function type comprises:

identifying one or more functional elements in the circuit design that have a particular level of independent operation from the remainder of the circuit design; and

defining a domain encompassing each identified functional element.

13. (Original) The method of claim 1 wherein each of the plurality of simulation domains provides a particular performance level and a particular resolution level, and wherein the particular simulation domains are selectively activated or deactivated during particular stages of simulation in combinations that either accelerate performance of the simulation environment or increase resolution of the simulation environment.

14. (Original) The method of claim 1 wherein selectively activating and deactivating the particular simulation domains comprises:

identifying a system of the circuit design;

determining which of the plurality of simulation domains are to be active for the identified system state; and

advancing simulation time only in each activated simulation domain.

15. (Original) The method of claim 14 wherein determining which of the plurality of simulation domains are to be active for the identified system state comprises at least one of a centralized control, a transaction-based control, and a distributed control.

16. (Original) The method of claim 15 wherein the centralized control comprises:

receiving the system state from one or more of the plurality of simulation domains;

consulting system configuration information to determine which of the plurality of simulation domains correspond to the particular system state; and

instructing a centralized simulation clock to advance only for those domains corresponding to the particular system state.

17. (Original) The method of claim 15 wherein the system state comprises system addresses in the circuit design.

18. (Original) The method of claim 15 wherein the system state comprises a data transaction in the circuit design, said data transaction being configured with information identifying which of the plurality of simulation domains are to be active for the data transaction, and wherein the transaction-based control comprises:

sending a message to a centralized simulation clock as part of the data transaction, said message to instruct the centralized simulation clock with respect to which of the plurality of simulation domains are to be active for the data transaction.

19. (Previously Presented) The method of claim 15 wherein a predetermined simulation domain is configured with activation information identifying at least one particular system state for which the predetermined simulation domain is to be active, wherein identifying the system state comprises receiving a broadcast of the system state at the predetermined simulation domain, and wherein distributed control at the predetermined simulation domain comprises:

determining if the predetermined simulation domain is to be active for the identified system state based on the activation information; and

advancing an operation in the predetermined simulation domain.

20. (Original) The method of claim 19 wherein the information further identifies an event for terminating operation of the predetermined simulation domain for the at least one particular system state.

21. (Original) The method of claim 14 wherein determining which of the plurality of simulation domains are to be active for the identified system state depends on a plurality of control mechanisms, wherein each of the plurality of control mechanisms comprises a priority level, and wherein a higher priority control mechanism takes precedence over a lower priority control mechanism.

22. (Original) The method of claim 1 wherein the plurality of simulation domains comprise a hierarchical structure, and wherein selectively activating and deactivating the particular simulation domains is based on levels of the hierarchical structure.

23. (Original) The method of claim 1 wherein each of the plurality of simulation domains comprises at least one simulation model, the method further comprising:

identifying state information comprising a transfer from a first simulation model in the simulation environment, said transfer being directed to a second simulation model in a circuit design being simulated in the simulation environment;

receiving the state information from the first simulation model; and

making the state information available to the second simulation model without simulating the transfer in the circuit design.

24. (Original) The method of claim 23 wherein simulating the transfer from the first simulation model to the second simulation model in the circuit design comprises transferring the state information through at least one additional simulation model in the simulation environment.

25. (Original) The method of claim 23 wherein receiving the state information and making the state information available comprises:

storing the state information in a coherent state memory space that is part of the simulation environment and corresponds to an element in the circuit design being simulated, said coherent state memory space being accessible to both the first simulation model and the second simulation model.

26. (Original) The method of claim 25 wherein the coherent state memory space is accessible to a plurality of additional simulation models.

27. (Original) The method of claim 23 wherein receiving the state information and making the state information available comprises at least one of:

a virtual transfer path for use when a simulation model of a transfer path in the circuit design is not included in the simulation environment; and

a higher performance transfer path than the simulation model of the transfer path in the circuit design.

28. (Original) The method of claim 27 wherein the higher performance transfer path provides a lower level of resolution than the simulation model of the transfer path in the circuit design.

29. (Original) The method of claim 1 wherein the first simulation model and a second simulation model of the plurality of simulation models represent different versions of a same functionality in the circuit design.

30. (Original) The method of claim 29 further comprising:

simulating the circuit design using the first simulation model, said first simulation model to generate state information; and

switching to simulate the circuit design using the second simulation model, said first simulation model to transfer the state information to the second simulation model prior to the second simulation model being used.

31. (Original) The method of claim 30 wherein the first simulation model and the second simulation model each have a particular level of performance and resolution, and wherein switching to the second simulation model is based on a change in a performance level and/or a resolution level desired at a different stage of simulation.

32. (Previously Presented) The method of claim 1 wherein a set of simulation models among a plurality of simulation models represent a same functionality in the circuit design, each of the set of simulation models having a particular level of performance and resolution, and each of the set of simulation models being used at different stages of simulation depending on a performance level and/or resolution level of the simulation.

33. (Previously Presented) The method of claim 25 wherein the simulation environment comprises a plurality of additional simulation models, each of the plurality of additional simulation models corresponding to one or more of a plurality of additional coherent state memory spaces, the method further comprising:

identifying additional state information comprising additional transfers among the plurality of additional simulation models in the simulation environment; and

storing the additional state information in ones of the plurality of additional coherent state memory spaces such that the additional state information is accessible to corresponding ones of the plurality of additional simulation models without simulating the additional transfers in the circuit design.

34. (Original) The method of claim 23 wherein both the first simulation model and the second simulation model are within a same simulation domain in the simulation environment.

35. (Original) The method of claim 23 wherein the first simulation model and the second simulation model are with different simulation domains in the simulation environment.

36. (Original) The method of claim 3 wherein the programming language simulator comprises at least one of a C programming language simulator, a C++ programming language simulator, a simulator using a C-based language, a simulator using a C++ based language, and a JAVA programming language simulator.

37. (Original) The method of claim 2 wherein the abstract model simulation domain comprises at least one of a hardware description language (HDL) simulator and a programming language simulator.

38. (Original) The method of claim 37 wherein the programming language simulator comprises at least one of a C programming language simulator, a C++ programming language simulator, a simulator using a C-based language, a simulator using a C++ based language, and a JAVA programming language simulator.

39. (Previously Presented) A machine readable storage medium having stored thereon machine readable instructions that when executed implement a method comprising:

selectively activating and deactivating particular simulation domains in a simulation environment such that a resolution and a performance for a circuit design being simulated can be dynamically modified; and

said simulation environment comprising a plurality of simulation domains.

40. (New) The machine readable storage medium of claim 39, wherein the method further comprises partitioning the circuit design into the plurality of simulation domains based on a partition criteria.